Hardware Security Implications of Reliability, Remanence and Recovery in Embedded Memory

Sergeis Skorobogatov
Department of Computer Science and Technology
University of Cambridge
Cambridge, UK
sps32@cam.ac.uk

***Abstract*—Secure semiconductor devices usually destroy key material on tamper detection. However, data remanence effect in SRAM and Flash/EEPROM makes secure erasure process more challenging. On the other hand, data integrity of the embedded memory is essential to mitigate fault attacks and Trojan malware. Data retention issues could influence the reliability of embedded systems. Some examples of such issues in industrial and automotive applications are presented. When it comes to the security of semiconductor devices, both data remanence and data retention issues could lead to possible data recovery by an attacker. This paper introduces a new power glitching technique that reduces the data remanence time in embedded SRAM from seconds to microseconds at almost no cost. This would definitely help in designing systems with better secret key guarding. Data remanence in non-volatile memory could be influenced in the same way. The effect of data remanence and data retention on hardware security is discussed and possible countermeasures are suggested. This should raise awareness among the designers of secure embedded systems.**

Keywords—data remanence; data retention; SRAM; EEPROM; Flash; glitching; hardware security; PUF; PRNG

I. INTRODUCTION

Semiconductor memory in the form of Static RAM (SRAM) was introduced in 1960s. It was later found to have data remanence problems similar to magnetic media with reliable data deletion [1,2,3,4]. Reprogrammable non-volatile memory (NVM) was first introduced as EEPROM in late 1970s and then as Flash in 1980s. They were also found to be affected by data remanence [5,6]. As a result, there is possibility that some information can still be extracted from previously erased memory. This could create problems with secure devices where designers assumed that all sensitive information is gone once the memory is erased. If the passwords or secret keys can be extracted afterwards, it could affect confidentiality of the previously encrypted information.

Reliability of data storage is paramount for modern embedded systems. Many people have come across situations when some microcontroller-based systems started behaving odd or stopped working. This might be home appliances, cars, industrial equipment etc. It seems that a serious reliability issue was overlooked and we might see more systems and devices starting to behave unpredictably or stop working. If it is a toaster or microwave oven you can cope, but what about old electronics equipment used in cars, aircrafts and industrial infrastructure? Previous research on car systems showed how the malfunction of certain car modules could pose a serious threat to passengers [7,8]. In this research possible cause of embedded systems sporadic failures was found and this could have very serious consequences.

In the 1980s, it was realised that low temperatures can increase the data retention time of SRAM to many seconds or even minutes. With the devices available at that time, it was found that increased data retention started at about −20°C and increased as temperature fell further [2]. Some devices are therefore designed with temperature sensors; any drop below −20°C is treated as a tampering event and results in immediate memory zeroisation [9,10]. The experiments described in this paper are set to measure the data remanence in modern microcontrollers to see if the low temperature data remanence still exists in embedded devices.

Usually the data remanence problem is tackled by wrapping the device into tamper protection enclosure with temperature sensors to prevent freezing and tamper sensors to detect intrusion [9,10,11]. This gives enough time to safely wipe secret key off the memory contents. However, designing a device with embedded SRAM having a low data remanence comes at a cost. This often requires modifying the SRAM cells to incorporate an additional destruction signal [12]. On practice this means that developers have very narrow choice of microcontrollers and system-on-chip (SoC) devices with such capability. For example, Maxim Integrated (former Dallas Semiconductor) offer secure microcontroller DS5250 with two self-destruct inputs which on activation wipe off the memory contents within microseconds [13]. Data remanence in NVM is usually defeated by several cycles of erasing and overwriting operations [6]. However, this cannot be accomplished in a short time especially for devices with a large memory size.

Chip manufacturers do not publish information about remanence effects in their chips. Of course, a developer can run data remanence tests on a selected batch of suitable chips. But this would take time and add cost to the design. The outcome of this research can be used to improve the protection of secure systems which use SRAM as a source of randomness and unique keys. First are used in pseudo random number generators (PRNG) employed in many cryptographic protocols.

Others are used to derive unique keys from physical unclonable functions (PUF). By reducing the data remanence time the contents of the SRAM can be refreshed more often and with higher randomness.

The research presented in this paper demonstrates that SRAM data remanence problem still exists in modern microcontrollers and like before deteriorates at low temperatures. This paper offers a low-cost solution to completely eliminate the data remanence problem at almost no cost. It also demonstrates that Flash/EEPROM can be affected in the same way and this could have influence on the security of embedded systems. Data retention can affect reliability of automotive and industrial systems. The results of this research should be of considerable concern to the developers of secure devices.

II. Background

Embedded systems are often based on microcontrollers – small integrated circuits with SRAM, ROM, EEPROM and Flash on a single silicon die.

Security engineers are interested in the period of time for which an SRAM device will retain data once the power has been removed. This is because many products do cryptographic and other security-related computations using secret keys or other variables that the equipment's operator must not be able to read out or alter. The usual solution is for the secret data to be kept in volatile memory inside a tamper-sensing enclosure. On detection of a tampering event, the volatile memory chips are powered down or even shorted to ground. If the data retention time exceeds the time required by an opponent to open the device and power up the memory, then the protection mechanisms can be defeated.

On power up the symmetric SRAM cells are tend to go more likely into one of the states 0 or 1. Each chip has unique and uncontrollable biases for each memory cell that depends on the intrinsic properties of semiconductor production. This helps to distinguish between each individual chip as well as use this for each device having unique PUF key for encryption. However, data remanence imposes some limitations on the time between each initialisation. Otherwise either random numbers will be predictable or the data used in PUF will always be the same making cloning much easier.

Another important thing to keep in mind is that security information could be restored even if part of the memory is corrupted. If an attacker has correctly restored only 90% of the 128-bit key he will have to search through $n!/(m!(n-m)!) = 128!/(115!13!) = 2.12\times10^{17} \sim 2^{58}$ possible keys. This is feasible to calculate in an hour with a medium-size cluster of computers or by using special hardware key cracker device. If only 80% of the information is known an attacker will need $2.51\times10^{26} \sim 2^{88}$ searches. Having even 100 times the capability, the attacker will spend more than a million years searching for the key. To simplify data remanence time calculations we assume that if 50% of information is lost then the key cannot be recovered anymore.

Unlike SRAM, which has only two stable logic states, EPROM, EEPROM and Flash cells store analog values in the form of a charge on the floating gate of a MOS transistor. The floating-gate charge shifts the threshold voltage of the cell transistor and this is detected with a sense amplifier when the cell is read.

Programmed floating-gate memories cannot store information forever. Up until late 90s these cells were not robust enough and usually held the charge for only 10–20 years. The failure mechanisms of EEPROM and Flash memory cells are well known. Various processes, such as field-assisted electron emission, de-trapping of electrons and ionic contamination, cause the floating gate to lose its charge, and this increases at higher temperatures [14]. Flash memory inherited similar data retention problems associated with the nature of data storage on a floating gate from the EEPROM. The failure rate doubles every ten degrees [15]. Another failure mode in the very thin tunnel oxides used in Flash memories is programming disturb, where unselected erased cells adjacent to selected cells gain charge when the selected cell is written. This is not enough to change the cell threshold sufficiently to upset a normal read operation, but could cause problems to the data retention time. Typical guaranteed data retention time for EPROM, EEPROM and Flash memories are 10, 40 and 100 years respectively. Those figures are usually refer to operating temperatures up to 85ºC. When devices are operating at higher temperatures, like in automotive applications up to 125ºC, the data retention time is likely to be reduced to just a few years.

There is a certain probability of distribution of weak cells which pass the initial testing after fabrication but cannot hold as much charge as normal cells. This could be caused by process variation during the fabrication, e.g. defect in material or contamination of masks or reagents. They could also be introduced by partial failures in the array control logic resulting in lower write voltage or shorter write time. These cells are likely to lose their charge and fail much faster especially during high temperature operation [16].

Failures of several embedded systems were investigated during this research. One was a failure in SCADA (supervisory control and data acquisition) [17] controllers used in the energy supply industry. Although company engineers were able to locate the cause of the system failure being the Motorola MC68HC11A1 microcontroller [18], they were unable to figure out why the microcontroller fails after certain time. It turned out that although hardware engineers did learn the specification and avoided possible data corruption problem, they were unable to predict that some other parameters could change during the device lifetime. These parameters are set at the chip factory after fabrication and testing and in most cases end users have no need to change them. Similar failures were detected in some car models where engine control unit stopped working after 8–10 years or car keys 'forgot' their encryption keys used for communication after 12 years. In those cases the cost of repair was substantial.

In many chips EEPROM and Flash cells could also be the part of configuration and security fuses. These fuses can control the size of available SRAM, EEPROM and Flash memory. In some microcontrollers the configuration is user accessible and even. In MC68HC11A1 the user can configure the OPTION register and select whether the ROM, EEPROM

and watchdog timer are enabled. If the ROM is disabled, the chip can only work with external memory. Usually chip manufacturers do not specify the data retention time for the chip configuration fuses despite to most of them being based on standard EEPROM cells. Hence, for system hardware engineer it would be natural to overlook the important parameters omitted by the manufacturer. The consequences could be devastating especially for high risk applications.

Memory technologies have significantly improved over the past 20 years. Most modern microcontrollers have guaranteed data retention time of over 40 years, but not all manufacturers specify the data retention parameters. Up until mid 2000s the microcontrollers with only 10 years of guaranteed data retention time were still manufactured. If they ended up in critical systems this could pose some serious problems. Although the mechanisms associated with data retention are well known and investigated, this does not eliminate the chances of failures caused by configuration memory especially as this memory cannot be easily tested in some cases.

When it comes to the hardware security, data recovery becomes the major concern. An attacker could potentially reverse engineer the embedded system by extracting all the information from embedded memory. Mask ROM is usually the easiest target because the information is stored in the form of present or absent transistor. This in some cases could be directly observed under optical microscope [19]. In other cases selective etching or microprobing would help [20,21]. Flash/EEPROM would require more sophisticated methods as the information is stored in the form of electrical charge. That means that either atomic force microscope (AFM) [22] or scanning electron micrposcope (SEM) [23] will be required. SRAM extraction is the most challenging task, because any interruption of the power supply could result in data loss. Also the switching energy of the memory cell is several orders of magnitude lower than in Flash/EEPROM. This makes SRAM the most secure storage. However, the requirement for having battery limits the areas of applications.

III. Experiments

As targets for the SRAM experiments the following microcontrollers were selected: Freescale (former Motorola) MC68HC908AZ60 [24] and MC68HC908AZ60A[25], Texas Instruments MSP430F112 [26] and MSP430F427 [27]. Several samples of each type were tested to measure the variation of data remanence time between devices within the same family. For the non-volatile memory experiments the following microcontrollers were selected: Microchip PIC16F873 [28], Atmel ATmega163 [29] and ATtiny12 [30], Motorola MC68HC11A1 and MC68HC11A8 [18]. Special test boards were built to efficiently communicate with the devices and test their embedded SRAM and Flash/EEPROM memory. The boards were connected to a PC with controlling software via RS-232 interface.

In the first set of experiments the chips were tested for SRAM data remanence at room temperature (+20ºC), low temperatures down to –30ºC and high temperatures up to +80ºC. For that each device was first initialised with data patterns in SRAM then powered down for different periods of time before powering it up and reading its memory contents. Both Motorola and Texas Instruments microcontrollers have on-chip monitor ROM which allows communication via RS-232 interface, hence, no special on-chip software is necessary to access embedded SRAM.

Previous research revealed that grounding power supply line to the device can significantly reduce data remanence time [4]. Therefore all used pins of the microcontroller were forced to GND at the start of measuring period. For temperature control a stacked Peltier elements with fan air cooling were used. For temperature monitoring a standard digital thermometer was used with external thermocouple and 0.1ºC resolution.

The next set of experiments with SRAM was aimed at verifying the new idea of how data remanence is affected by power glitching. Instead of a smooth power down process on the VCC line (Figure 1, blue trace), the switching was performed with deliberate overshooting (orange trace). This was achieved by designing a special power supply unit controlled by a PC where the power supply voltage was sampled by digital-to-analog converter. The overshooting glitch was formed by an operational amplifier with capacitive load. Then the signal was buffered to supply enough current to the device under test. The amplitude of the glitch was regulated on the test board using passive LCR low pass filters.

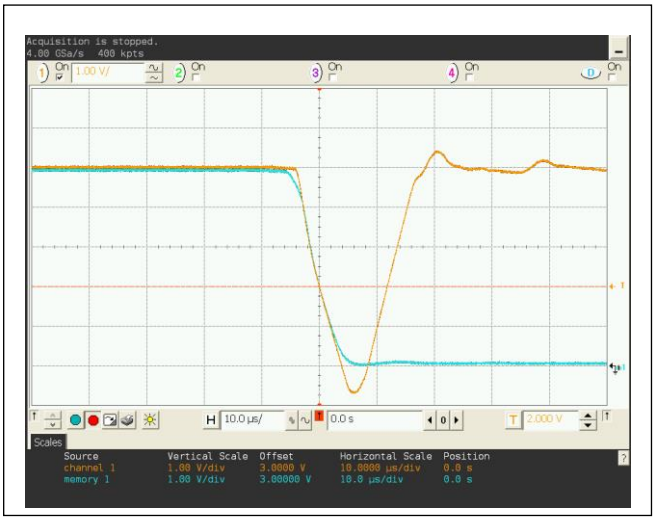

Fig. 1. Power down process on VCC line: blue – normal, orange – glitch

The same glitching experiments were repeated for the process of secure erasure of Flash/EEPROM in Microchip and Atmel microcontrollers. Chip manufacturers claim that the process of chip erase is performed in a way that the information from code and data memory is removed well before the security fuses are reset. The glitching could have effect on that process, like it had on SRAM data remanence. The microcontrollers were programmed with a test pattern then secured by setting their security fuses. The power glitches of different duration and amplitude were applied before starting the chip erase operation. Then after 100ms time the chip was

powered down for 1 second before checking the state of its memory and fuses.

Another experiment was carried out on a partially working old optical microscope Leitz Ergolux. The electronic board inside its controller was behaving odd thus sometimes moving the stage and lenses unexpectedly. Inside the controller was a microprocessor-based board. The problem was rectified by reading the 2764 EPROM IC at different voltages and then reprogramming the same memory chip. Similar issues with EPROM and EEPROM reliability were found in many old cars. Car repair garages have to deal with many old electronic blocks in cars which failed after some loss of data. Refreshing the embedded memory inside microcontrollers solved the problem in 90% cases.

The final set of experiments was performed to measure data retention time in Motorola MC68HC11Ax microcontrollers. For that a UV light was used as an ageing source to establish how the memory cells lose their charge and what effect this has on the chip operation. To access the die the sample of MC68HC11A1 chip was decapsulated with standard process using fuming nitric acid [20]. The UV erasure experiments were aimed at speeding up the ageing process to observe how the memory cells lose their charge and any correlation with OPTION register.

IV. Results

Data remanence results at room temperature for four microcontrollers are presented in Figure 2. The time coordinate is in logarithmic scale as this helps to see the transition clearer. The older Motorola microcontroller MC68HC908AZ60 has the longest data remanence time, while Texas Instrument microcontroller MSP430F112 has the shortest time. Figure 3 shows distribution of data remanence time between nine different samples of similar devices – three of MC68HC908AZ60, three of MC68HC908AZ60A (mask 2J74Y) and three of MC68HC908AZ60A (mask 3K85K). As it can be observed, differences between data remanence time of samples from the same batch could be larger than between average samples of different devices.

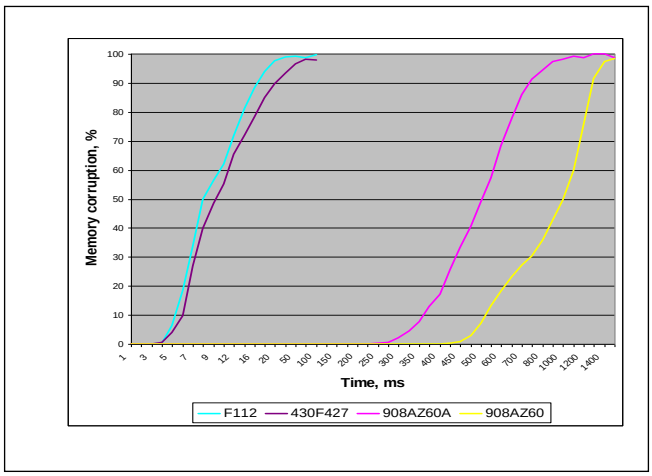

Fig. 2. Data remanence effect in different microcontrollers

At low temperature the data remanence time is expectedly increased (Figure 4) for both Motorola and Texas Instruments microcontrollers. The time was counted to the moment when 50% of cells lost their state. The data remanence time is accelerated at temperatures below 0ºC. This leads to seconds for MSP430F112 and minutes for MC68HC908AZ60A at –30ºC. At higher temperature the data remanence time is reduced to milliseconds for MC68HC908AZ60A at +80ºC. In the logarithmic scale of Figure 4 the temperature dependency of data remanence is almost linear.

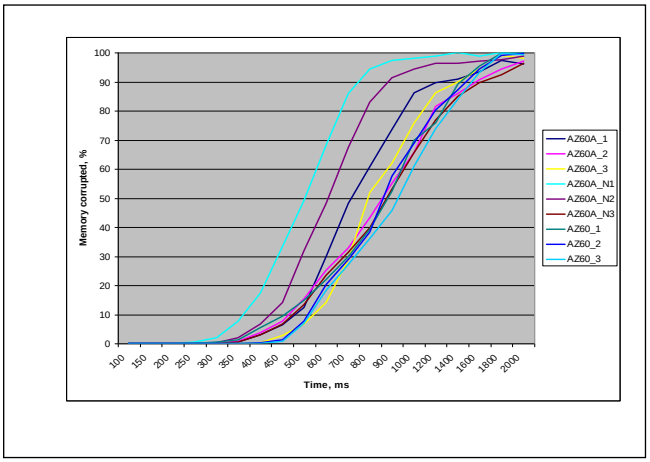

Fig. 3. Data remanence difference between batches

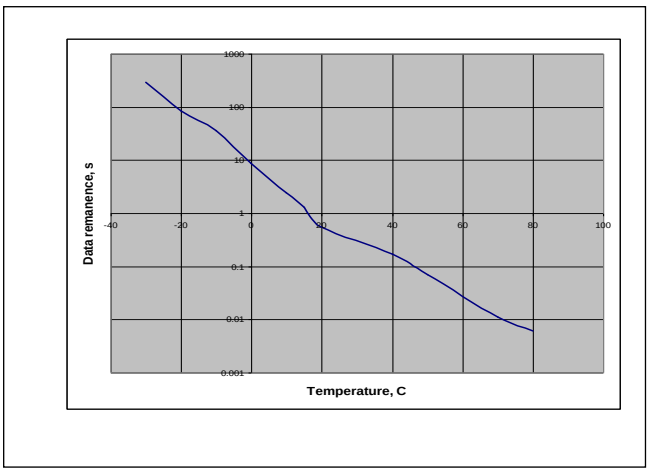

Fig. 4. Temperature dependence of data remanence time

The effect of power glitching on the data remanence time was very significant for all samples of tested microcontrollers. For all microcontrollers the memory contents was completely wiped off to a random state with as short as 5µs glitch (Figure 1). Even for the MSP430F112 microcontroller which normally has about 8ms data remanence time at room temperature this is a thousand times improvement. While for MC68HC908AZ60A microncontroller with half a second data remenence time it is significantly better. At low temperatures the glitching helps with the same efficiency – the data remanence time reduced from several minutes to a few

microseconds. This effect is likely to be caused by a quick discharge of SRAM cells transistors into negatively biased power line. At the same time the short and controlled glitch pulse does not cause a latch-up effect in CMOS transistors.

Power glitching of EEPROM and Flash microcontrollers was different from the SRAM. For all tested samples of Microchip PIC16F873, and Atmel ATmega163 and ATtiny12, the glitching caused data retention of the main EEPROM and Flash memory to increase at a certain threshold. At the same time the security fuses were unaffected. This resulted in the security of those microcontrollers being circumvented once the glitching pulse was large enough. Because the chip erase process is internally timed, it is not possible to erase the memory forever. If it is not erased within 20ms the control circuit shuts down the erasure process.

The final set of experiments was carried out on decapsulated Motorola MC68HC11A1 microcontroller for the purpose of finding out if the data retention time of the configuration fuses is the same as for the on-chip EEPROM. The chip manufacturer specifies EEPROM data retention time as at least 10 years, however, it does not specify that time for the EEPROM bits of OPTION register.

Close look at the die revealed all important areas in the embedded EEPROM. Partial reverse engineering of the memory physical map was carried out using semi-invasive methods [19]. For that different parts of the EEPROM array on the decapsulated sample were exposed to UV light after programming the memory. Writing 0's into the memory causes the cells' transistors to store the charge. When the charge is removed with UV light the memory cell is read as '1' or erased state. One of the EEPROM blocks has extra row line and it turned out that the part of it is used for storing the OPTION register. The additional EEPROM cells look exactly the same as the normal EEPROM memory cells. Therefore, their behavior and parameters should remain the same. In order to compare the data retention time of those special cells with normal cells, the dependency of the data retention from the UV exposure time was measured. The time when half of the cells have lost their charge was 45 minutes and was identical for both areas. This suggests that the data retention parameters of the EEPROM could be used to estimate the guaranteed retention time for the OPTION register.

The consequence of unintentional change of the OPTION register contents could be devastating. This is because this register controls the presence of ROM and EEPROM in the memory map. While the EEPROM corruption could be mitigated using error correction techniques, sudden change of the memory layout will likely cause the system to crash.

V. Conclusion

The research presented in this paper showed that data remanence still exists in modern microcontrollers. For some devices the effect could be very serious and allow to keep the secret information and keys for several seconds at room temperature. At low temperature down to –30ºC it could increase to minutes. Previous solutions were either too expensive (special SRAM cells) or bulky (tamper enclosure with temperature sensors). Heating up the chip might help a bit, but might not be suitable due to excessive heat and bulky design. This paper introduces a solution to that problem which reduces the data remanence time to a few microseconds thus completely eliminating the effect. The cost of the proposed solution is very low as it is completely non-invasive. The method of significant reduction of data remanence time can be used for SRAM based PUF and PRNG devices in order to speed up the process of key generation.

Data remanence time varies between different families of microcontrollers and between devices within each family because of variation between transistors. Therefore, secure system designers should test wide range of samples to ensure their robustness against data remanence at wide range of conditions for all samples.

Data remanence in non-volatile memory could be influenced as well. However, in that case it could have severe effect on security. It might be possible to increase data retention time of the main Flash/EEPROM memory in a way that it would not lose the data fast enough during mass erase used for secure reprogramming. As a result the protection fuse will be disabled earlier than expected thus leaving the on-chip memory contents intact. This would compromise the security of the chip. Modern microcontrollers should be properly tested by the manufacturers to avoid such situations where protection fuse is erased earlier than the main memory as a result of power glitching.

Interconnection between data retention and data remanence was discussed with real examples in automotive and industrial systems. This research has demonstrated that old semiconductor devices with embedded EEPROM and Flash memory could pose serious reliability issues by starting malfunctioning after 10–20 years. For some devices the system design engineers could be completely unaware about possible problems because the chip manufacturer might not specify parameters of the EEPROM cells used for internal factory debug and testing purposes. However, when those cells change their state through normal ageing process the device operation will be affected. Although ageing of microcontrollers is a concern for industry, it was mainly investigated for space applications rather than automotive and industrial ones. Also, none of those investigations were aimed at security aspects but rather on software integrity, data storage and limited number of reprogramming cycles for Flash/EEPROM storage.

The danger of widely used microcontroller-based systems going into sporadic failures is hard to overestimate. Electronic modules in old cars sometime start to play up, however, this is often being treated as normal ageing failure, hence, the owner or insurance company pays the bill. However, the underlying problem is much deeper. The outcome and cause of the problem could be devastating for other industries as it lies at intersection between hardware, software, reliability and security. For most chips manufacturers guarantee at least 40 years of data retention time. However, if chips are operating at higher temperatures, like in automotive applications, their actual data retention time could be much shorter.

There are several ways how the retention time of the special cells can be improved. For example, multiple cells can be used to reduce the probability of a failure caused by a single cell.

Another way is to design special cells which lose charge slower than normal ones. Also, the devices should be better tested for the use in critical applications.

Latest SoCs and microcontrollers have extended lifetime of embedded NVM. However, emerging memory technologies could have reliability issues. More robust testing will be required to make sure that automotive parts will still be fully functional after 20 years in adverse conditions.

There are other implications from undocumented features present in many chips and mainly used for factory failure analysis and debugging purposes. In this case the hardware design engineers will be unaware of possible problems and outcomes unless they could afford to scan the device for undocumented features [31].

The results presented in this paper were based on mid-range 8-bit and 16-bit microcontrollers built with 0.8µm to 0.25µm process. It would be beneficial to expand it with measurements applied to higher density devices down to 90nm process and beyond. Those devices are likely to have lower data remanence due to higher leakage in SRAM cells. However, lower operating voltage could compensate this making them still vulnerable to data remanence.

More robust testing and evaluation must be performed on semiconductor devices going into sensitive applications with high risk factors like in car, aviation and medical industries as well as critical infrastructure. For those systems which remain in the field it would be beneficial to carry out risk assessment and reprogram or upgrade the hardware.

REFERENCES

[1] A Guide to Understanding Data Remanence in Automated Information Systems. Version 2, September 1991, NSA/NCSC Rainbow Series

[2] Peter Gutmann: Secure Deletion of Data from Magnetic and Solid-State Memory. In Proceedings of 6th USENIX Security Symposium, San Jose, California, July 1996, pp 77–89

[3] Ross J. Anderson, Markus G. Kuhn: Tamper Resistance – a Cautionary Note. The Second USENIX Workshop on Electronic Commerce, Oakland, California, November 1996

[4] Sergei Skorobogatov: Low Temperature Data Remanence in Static RAM. Technical Report UCAM-CL-TR-536, University of Cambridge, Computer Laboratory, June 2002

[5] Peter Gutmann: Data Remanence in Semiconductor Devices. 10th USENIX Security Symposium, Washington, D.C., August 13–17, 2001

[6] Sergei Skorobogatov: Data Remanence in Flash Memory Devices. Cryptographic Hardware and Embedded Systems Workshop (CHES), LNCS 3659, Springer-Verlag, 2005, ISBN 3-540-28474-5, pp 339–353

[7] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage: Experimental Security Analysis of a Modern Automobile. IEEE Symposium on Security and Privacy, Oakland, CA, 16–19 May 2010

[8] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage: Comprehensive Experimental Analyses of Automotive Attack Surfaces. USENIX Security, 10–12 August 2011

[9] Steve H. Weingart: Physical Security for the mABYSS System. In proceedings of the IEEE Computer Society Conference on Security and Privacy, 1987, pp. 52–58

[10] Sean W. Smith, Steve Weingart: Building a High-Performance, Programmable Secure Coprocessor. Computer Networks 31, April 1999, pp. 831–860

[11] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses, Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000), Springer-Verlag LNCS 1965, pp. 302–317

[12] Yu Kai, Zou Xuecheng, Yu Guoyi and Wang Weixu: Security strategy of powered-off SRAM for resisting physical attack to data remanence. Journal of Semiconductors, Vol. 30, No. 9, September 2009

[13] DS5250 High-Speed Secure Microcontroller.
http://www.maximintegrated.com/en/products/digital/microcontrollers/DS5250.html

[14] S. Aritome, R. Shirota, G. Hemink, T. Endoh, F. Masuoka: Reliability issues of flash memory cells. Proceedings of the IEEE, Vol. 81, No. 5, May 1993, pp 776–788

[15] P. Lecuyer: Reliability prediction for microcontrollers with embedded EEPROM. Proceedings of the European Space Components Conference, ESCCON 2002, 24–27 September 2002, Toulouse, France, ISBN 92-9092-817-4, pp 67–71

[16] Y. Chen, R. Kemski, L. Scheick, F. Stott, D. Nguyen, T. Nguyen, R. Bennett, K. Erickson: EEPROM bit failure investigation. 6th Military and Aerospace Programmable Logic Devices International Conference, Washington D.C., USA, 9 September 2003

[17] SCADA.
http://en.wikipedia.org/wiki/SCADA

[18] MC68HC11A8 HCMOS Single-Chip Microcontroller, Freescale Semiconductor Inc.
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC11A8.pdf

[19] Sergei Skorobogatov, Semi-invasive attacks – A new approach to hardware security analysis, Technical Report UCAM-CL-TR-630, University of Cambridge, Computer Laboratory, April 2005.

[20] O. Kömmerling, M.G. Kuhn: Design principles for tamper-resistant smartcard processors. USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 1999

[21] S. Skorobogatov: How microprobing can attack encrypted memory. In Proceedings of Euromicro Conference on Digital System Design, AHSA 2017 Special Session, Vienna, Austria. IEEE Computer Society, 2017

[22] C. De Nardi, R. Desplats, P. Perdu, F. Beaudouin and J.-L. Gauffier, Oxide charge measurements in EEPROM devices, Microelectronics Reliability, Vol.45, 2005, pp 1514-1519

[23] Franck Courbon, Sergei Skorobogatov, Christopher Woods: Reverse Engineering Flash EEPROM Memories Using Scanning Electron Microscopy. Smart Card Research and Advanced Applications (CARDIS 2016), LNCS vol 10146, Springer, 2017

[24] MC68HC908AZ60 HCMOS Microcontroller Unit, Freescale Semiconductor Inc.
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908AZ60.pdf

[25] MC68HC908AZ60A Microcontroller, Freescale Semiconductor Inc.
http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908AZ60A.pdf

[26] MSP430F112 Ultra-Low-Power Microcontroller, Texas Instruments.
http://www.ti.com/product/msp430f112

[27] MSP430F427 Ultra-Low-Power Microcontroller, Texas Instruments.
http://www.ti.com/product/msp430f427

[28] PIC16F87X Data Sheet. 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, Microchip.

[29] Atmega163, Atmel.
http://www.atmel.com/devices/atmega163.aspx6

[30] Attiny12, Atmel.
http://www.atmel.com/devices/attiny12.aspx

[31] S. Skorobogatov, C. Woods. Breakthrough silicon scanning discovers backdoor in military chip. Cryptographic Hardware and Embedded Systems Workshop (CHES-2012), 9–12 September 2012, LNCS 7428, Springer, ISBN 978-3-642-33026-1, pp 23–40